

Solución ejercicio propuesto por el profesor:

Esto es lo que debemos escribir en el script

```
u=c(-10,exp(1),sin(2*pi/3),-5,4/3)
v=c(3,25,10^3,-50,sqrt(13))
u
v
A=matrix(c(5/3,10^2,88,factorial(4),exp(2),0.5,99,3,cos(3*pi/5),-2,0.25,-2,10,sqrt(6),sqrt(13),0),nrow=4,ncol=4)
A
B=matrix(c(-6,-3/8,sqrt(88),factorial(8),12,1,pi,2,cos(3*pi),1,-51,-1,tan(8),88,log(6),14),nrow=4,ncol=4)
B
u+v
u-v
u%*%v
C<-cbind(u,v)
C
R=rbind(u,v)
R
A+B
A%*%B
A*B
b<-matrix(c(2,1,-1,3),nrow=1,ncol=4)
b
solve(A,b)
Al=solve(A)
Al
BT=t(B)
BT
A%*%BT
eigen(BT)
```

Explicación de lo que aparece escrito en el script:

1. Tenemos que definir los vectores u y v .
2. Leemos los vectores u y v para asegurarnos de que no haya ningún error.
3. Tenemos que definir las matrices A y B , para ello debemos utilizar el comando **matrix** tal y como aparece en el script.
4. Leemos las matrices definidas anteriormente para asegurarnos de que no haya ningún error.
5. Tenemos que sumar los vectores u y v , para ello debemos utilizar el operador **+**; ($u+v$)
6. Tenemos que restar los vectores u y v , para ello debemos utilizar el operador **-**; ($u-v$)
7. Tenemos que realizar el producto escalar, para ello debemos utilizar el operador **%*%**; ($u\%*\%v$).
8. Debemos construir una matriz C cuyas columnas sean los vectores u y v , para ello utilizamos el comando **cbind(u,v)** y a continuación la leemos para asegurarnos de que no haya ningún error.
9. Debemos construir una matriz R cuyas filas sean los vectores u y v , para ello utilizamos el comando **rbind(u,v)** y a continuación la leemos para asegurarnos de que no haya ningún error.
10. Tenemos que sumar las matrices A y B , para ello debemos utilizar el operador **+**; ($A+B$).
11. Tenemos que multiplicar las matrices A y B , para ello debemos utilizar el operador **%*%**; ($A\%*\%B$)
12. Tenemos que multiplicar las matrices A y B elemento a elemento para ello debemos utilizar el operador *****; ($A*B$).
13. Para poder resolver el sistema de ecuaciones que aparece en el enunciado, debemos utilizar el vector $b=(2,1,-1,3)$ para construir la matriz de términos independientes.

14. Resolvemos el sistema de ecuaciones, para ello debemos utilizar el comando **`solve(A,b)`**.
15. Tenemos que calcular la matriz inversa de A, para ello debemos utilizar el comando **`solve(A)`**; (`A1=solve(A)`).
16. Tenemos que leer A1 para asegurarnos de que no haya ningún error.
17. Tenemos que calcular la matriz transpuesta de B, para ello debemos utilizar el comando **`t(B)`**; (`BT=t(B)`).
18. Tenemos que leer BT para asegurarnos de que no haya ningún error.
19. Tenemos que realizar el producto de las matrices A y B, para ello debemos utilizar el operador **`%*%`**; (`A*%BT`).
20. Por último, tenemos que obtener los valores propios de la matriz BT, para ello debemos utilizar el comando **`eigen()`**; (`eigen(BT)`).

Una vez hemos escrito todo esto en el script y lo ejecutamos, la solución del problema será la que aparece en la consola de la fotografía que aparece a continuación.

Esto es lo que aparece en la consola al ejecutar el script, que es la solución del problema:

```
RGui (64-bit) - [R Console]
Archivo Editar Visualizar Misc Paquetes Ventanas Ayuda

[1] u=c(-10,exp(1),sin(2*pi/3),-5,4/3)
[2] v=c(3,25,10^3,-50,sqrt(13))
[3] u
[1,] -10.00000000  2.7182818  0.8660254 -5.0000000  1.3333333
[4] v
[1,]  3.0000000  25.0000000 1000.0000000 -50.0000000  3.605551
[5] A=matrix(c(5/3,10^2,88,factorial(4),exp(2),0.5,99,3,cos(3*pi/5),-2,0.25,-2,10,sqrt(6),sqrt(13),0),nrow=4,ncol=4)
[6] A
      [,1] [,2] [,3] [,4]
[1,]  1.666667  7.389056 -0.309017 10.000000
[2,] 100.000000  0.500000 -2.000000  2.449490
[3,]  88.000000  99.000000  0.250000  3.605551
[4,]  24.000000  3.000000 -2.000000  0.000000
[7] B=matrix(c(-6,-3/8,sqrt(88),factorial(8),12,1,pi,2,cos(3*pi),1,-5i,-1,tan(8),88,log(6),14),nrow=4,ncol=4)
[8] B
      [,1] [,2] [,3] [,4]
[1,] -6.000000 12.000000 -1 -6.799711
[2,] -0.375000  1.000000  1  88.000000
[3,]  9.380832  3.141593 -51  1.791759
[4,] 40320.000000  2.000000 -1 14.000000
[9] u+v
[1,] -7.000000  27.718282 1000.866025 -55.000000  4.938885
[10] u-v
[1,] -13.000000 -22.281718 -999.133975  45.000000 -2.272218
[11] u%*%v
      [,1]
[1,] 1158.79
[12] C<-cbind(u,v)
[13] C
      u      v
[1,] -10.00000000  3.000000
[2,]  2.7182818  25.000000
[3,]  0.8660254 1000.000000
[4,] -5.00000000 -50.000000
[5,]  1.33333333  3.605551
[14] R=rbind(u,v)
[15] R
      [,1] [,2] [,3] [,4] [,5]
u -10  2.718282  0.8660254 -5  1.333333
v  3  25.000000 1000.0000000 -50  3.605551
[16] A+B
      [,1] [,2] [,3] [,4]
[1,] -4.333333 19.38906 -1.309017  3.200289
[2,]  99.625000  1.50000 -1.000000  90.449490
[3,]  97.380832 102.14159 -50.750000  5.397311
[4,] 40344.000000  5.00000 -3.000000 14.000000
[17] A%*%B
      [,1] [,2] [,3] [,4]
[1,] 403184.3303  46.41825 11.48225615  778.35040
[2,]  98144.4773 1199.11579  0.05051026 -605.26181
[3,] 144813.0476 1162.99650 -5.35555128  8164.55105
[4,] -163.8867  284.71681  81.00000000  97.22341
[18] A*B
      [,1] [,2] [,3] [,4]
[1,] -10.0000  88.66867  0.309017 -67.997115
[2,] -37.5000  0.50000 -2.000000 215.555097
[3,]  825.5132 311.01767 -12.750000  6.460281
[4,] 967680.0000  6.00000  2.000000  0.000000
```

```

> b<-matrix(c(2,1,-1,3),nrow=1,ncol=4)
> b
      [,1] [,2] [,3] [,4]
[1,]    2    1   -1    3
> solve(A,b)
Error in solve.default(A, b) :
  'b' (1 x 4) debe ser compatible con 'a' (4 x 4)
> A1=solve(A)
> A1
      [,1]      [,2]      [,3]      [,4]
[1,] -0.0032379120  0.01242641  0.0005382671 -0.01185884
[2,] -0.0006556883 -0.01176687  0.0098125667  0.01309475
[3,] -0.0398384769  0.13146665  0.0211780550 -0.62266401
[4,]  0.0997930671  0.01068608 -0.0066858338 -0.02694069
> BT=t(B)
> BT
      [,1] [,2]      [,3] [,4]
[1,] -6.000000 -0.375  9.380832 40320
[2,] 12.000000  1.000  3.141593    2
[3,] -1.000000  1.000 -51.000000   -1
[4,] -6.799711 88.000  1.791759   14
> A%*%BT
      [,1]      [,2]      [,3]      [,4]
[1,]  10.98058 886.4550  72.52558  67355.09
[2,] -608.65582 176.5551 1046.04284 4032037.29
[3,]  635.23329 383.5385 1130.24113 3548408.23
[4,] -106.00000 -8.0000  336.56473  967688.00
> eigen(BT)
eigen() decomposition
$values
[1] -68.30979+537.7113i -68.30979-537.7113i 145.51944+ 0.0000i
[4] -50.89987+ 0.0000i

$vectors
      [,1]      [,2]      [,3]
[1,]  0.99966350+0.000000000i  0.99966350+0.000000000i  0.996556651+0i
[2,] -0.00276842-0.021946675i -0.00276842+0.021946675i  0.082698250+0i
[3,] -0.00000568+0.001861566i -0.00000568-0.001861566i -0.004669285+0i
[4,] -0.00154489+0.013330970i -0.00154489-0.013330970i  0.003746838+0i
      [,4]
[1,] -1.292653e-01+0i
[2,] -3.010492e-02+0i
[3,]  9.911529e-01+0i
[4,] -8.693285e-05+0i

```